Atomic Architecture

An Architectural Vision for Better Information Systems

Malcolm Sparks, CTO@JUXT

JUXT

Who am I?

Malcolm Sparks, CTO@JUXT

@malcolmsparks (GitHub, X, LinkedIn, etc.)

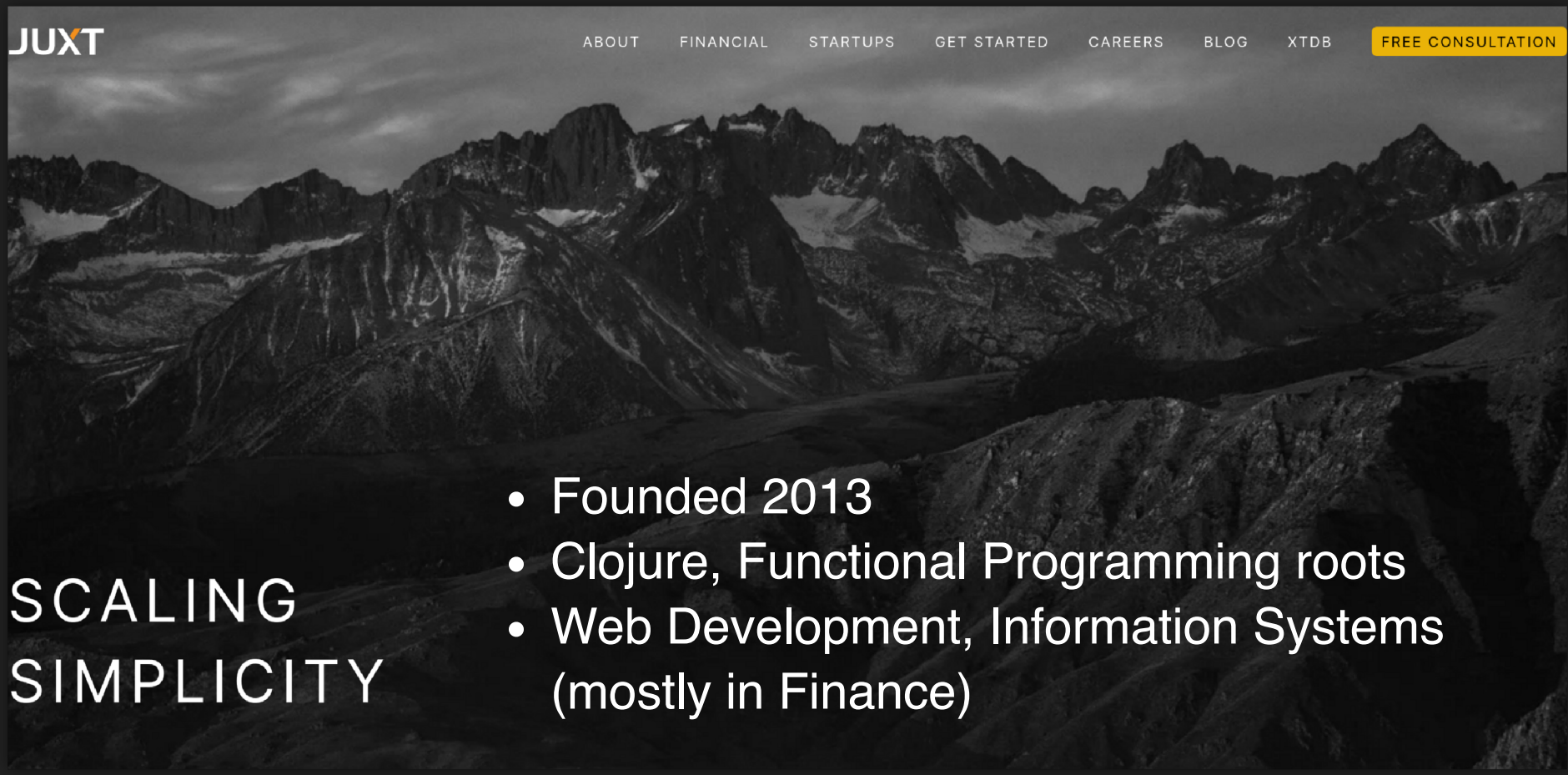
Languages

1982-1991	BASIC, 68000
1991-1996	C, C++, UNIX
1996-2009	Java, XML
2009-2023	Clojure

Who am I?

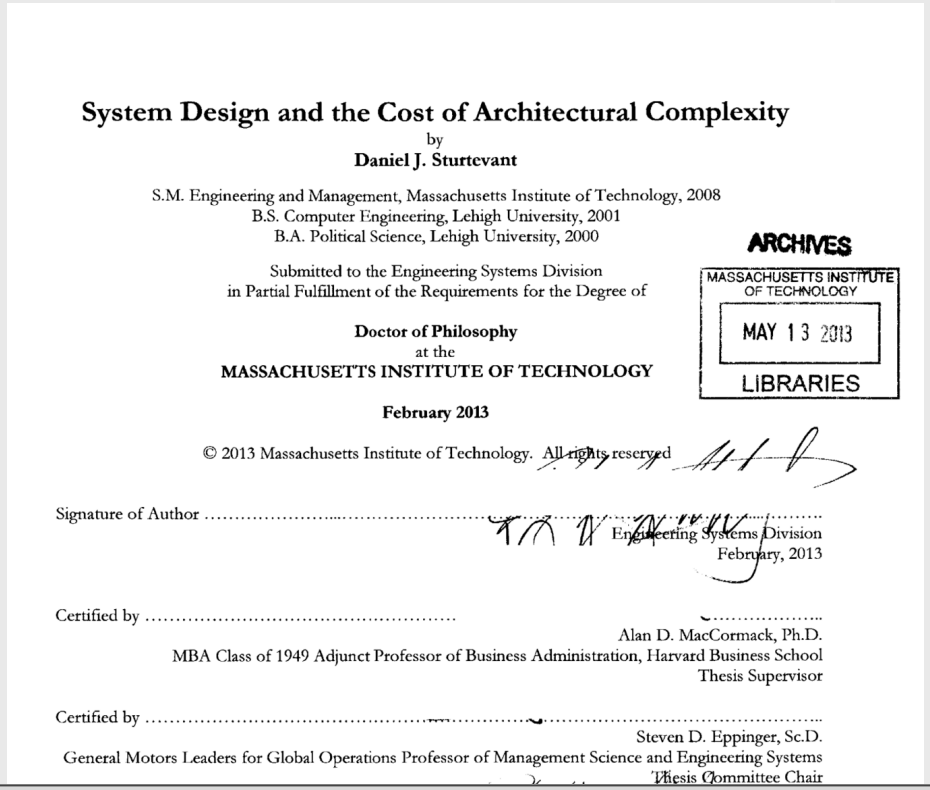
Projects

1997	Server-side Java, CORBA, RMI, Servlets
1998	Enterprise JavaBeans
1999-2003	J2EE App Servers, Servlet/JSP containers
2006+	REST (Restlet, WebMachine)
2010	plugboard, Liberator
2014	bidi, yada
2019+	jinx, pick, reap, tick, grab...
2020-2023	Site (part of XTDB family)


JUXT
ABOUT
FINANCIAL
STARTUPS
GET STARTED
CAREERS
BLOG
XTDB
FREE CONSULTATION
SCALING
SIMPLICITY
• Founded 2013
• Clojure, Functional Programming roots
• Web Development, Information Systems (mostly in Finance)

JUXT

Motivation
Status Quo



System Design and the Cost of Architectural Complexity

by

Daniel J. Sturtevant

S.M. Engineering and Management, Massachusetts Institute of Technology, 2008
B.S. Computer Engineering, Lehigh University, 2001
B.A. Political Science, Lehigh University, 2000

Submitted to the Engineering Systems Division
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2013

Signature of Author .....
Engineering Systems Division
February, 2013

Certified by .....
Alan D. MacCormack, Ph.D.
MBA Class of 1949 Adjunct Professor of Business Administration, Harvard Business School
Thesis Supervisor

Certified by .....
Steven D. Eppinger, Sc.D.
General Motors Leaders for Global Operations Professor of Management Science and Engineering Systems
Thesis Committee Chair

“ Within this research setting, we found that differences in architectural complexity could account for 50% drops in productivity, three-fold increases in defect density, and order-of-magnitude increases in staff turnover.

Source: System design and the cost of architectural complexity, **Sturtevant (2013)**

https://dspace.mit.edu/handle/1721.1/79551

Principle 1	**Shared State**
Principle 2	Domain Operations
Principle 3	Data Consistency
Principle 4	API Only
Principle 5	Access Control
Principle 6	Event Logging
Principle 7	Bitemporality

Loosely coupled monoliths and where to find them

Andras Gerlits · Follow

Published in ITNEXT · 6 min read · Nov 23, 2022

“ For the last decade, the tech sector has been on a mission to decrease its dependence on central SQL-monoliths. The mainstay of these efforts were focused on microservices and its “loose coupling” and “separation of concerns” as guiding principles.

https://itnext.io/loosely-coupled-monoliths-and-where-to-find-them-4004fac8ecc1

Microservices and the myth of loose coupling

Andras Gerlits · Follow

Published in ITNEXT · 6 min read · Oct 27, 2021

35 2

*"the issue comes from the interaction between two processes happening in parallel. Remember that the point of this (increasingly complex) exercise is to **decrease coupling**, ie: reduce the amount of implied information between subsystems.*

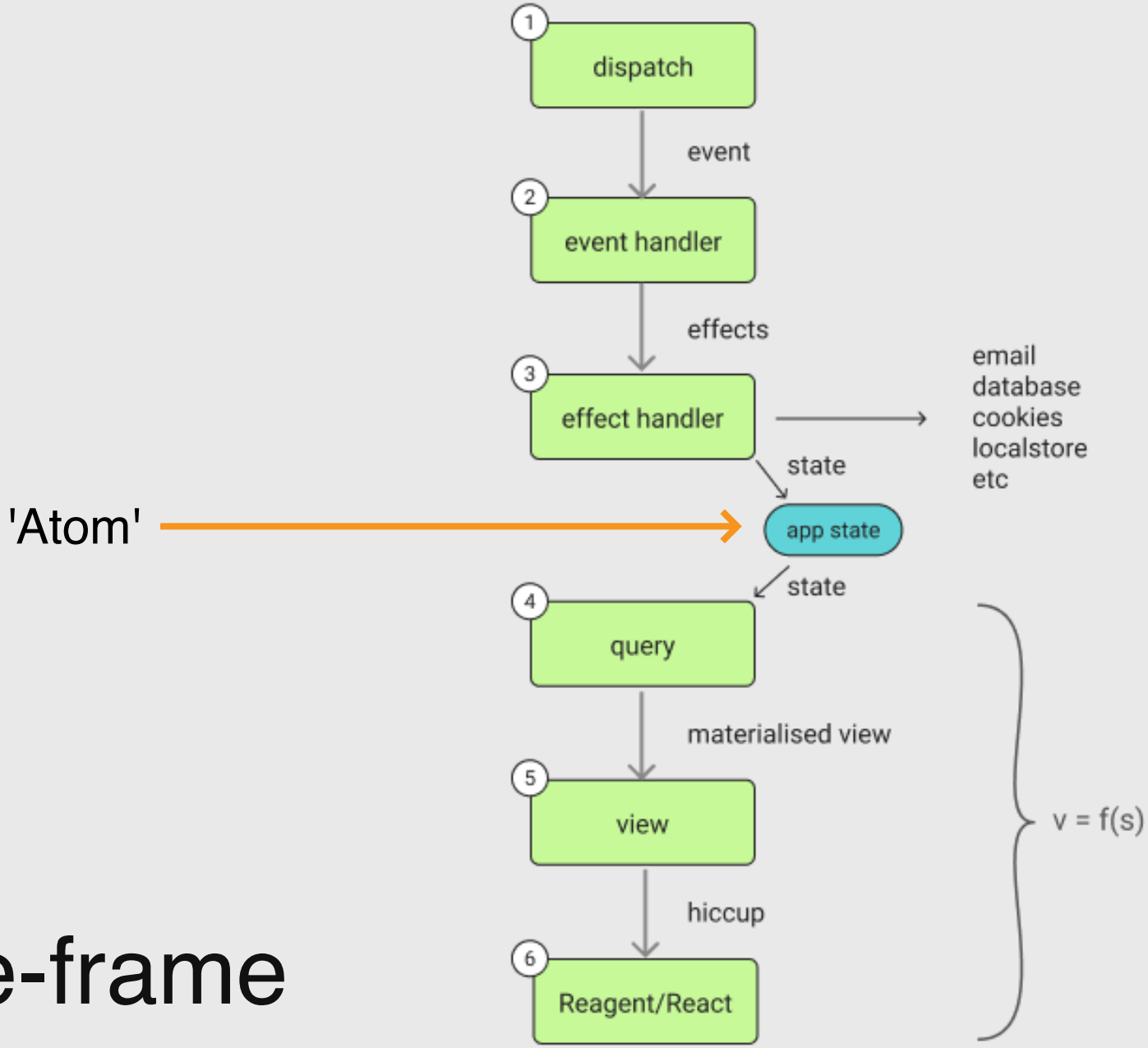
https://itnext.io/microservices-and-the-myth-of-loose-coupling-9bbca007ac1a

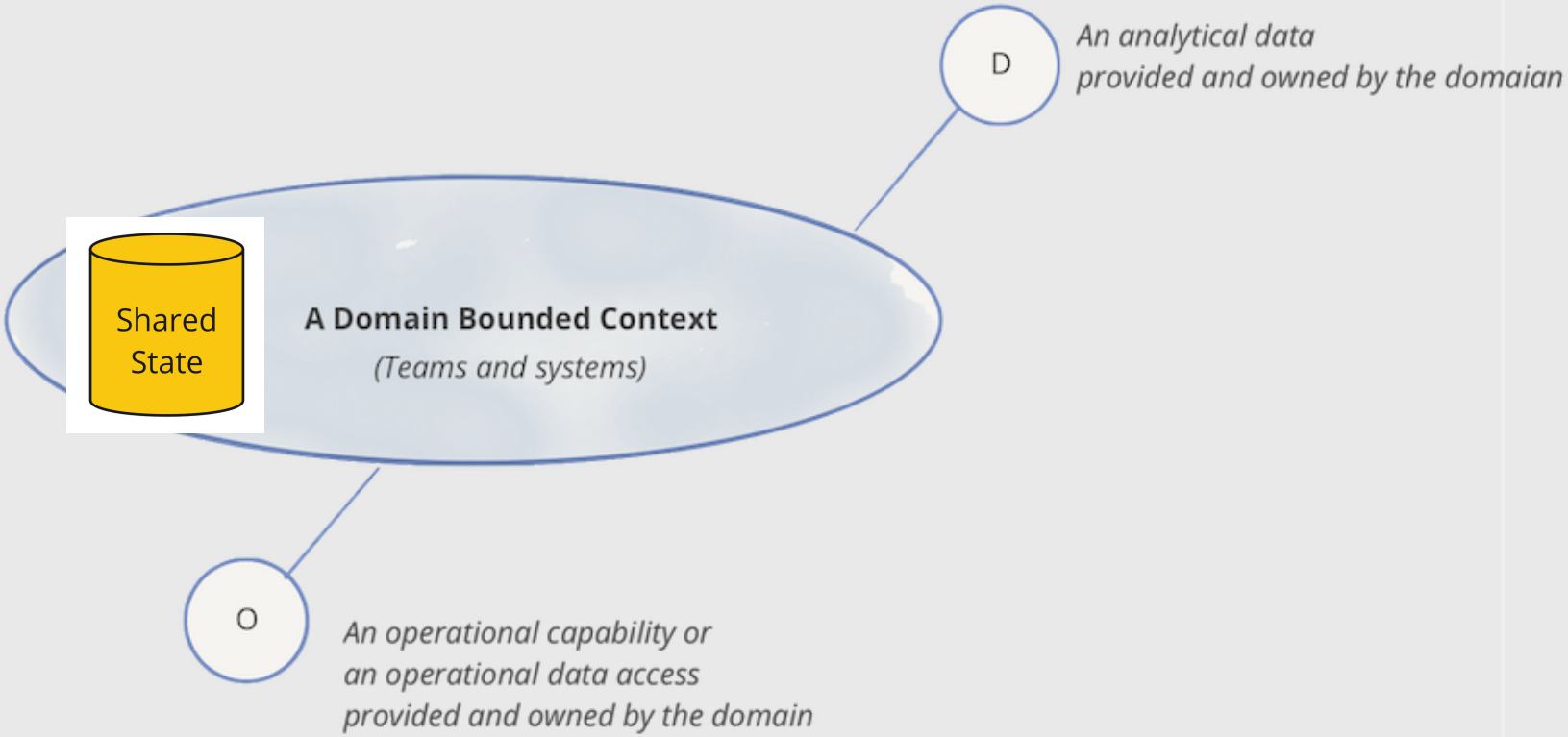
Scripting

```
some_list = [10, 20, 30, 40]


print("Original List:", some_list)
some_list.append(100)
some_list[0] = 0
some_list.pop(1)
print("Manipulated List:", some_list)
```

re-frame


1
dispatch
event
2
event handler
effects
3
effect handler
email
database
cookies
localstore
etc
state
'Atom'
app state
state
4
query
materialised view
5
view
v = f(s)
hiccup
6
Reagent/React



from https://martinfowler.com/articles/data-mesh-principles.html

Centralized schemas and microservices, match made in Hell?

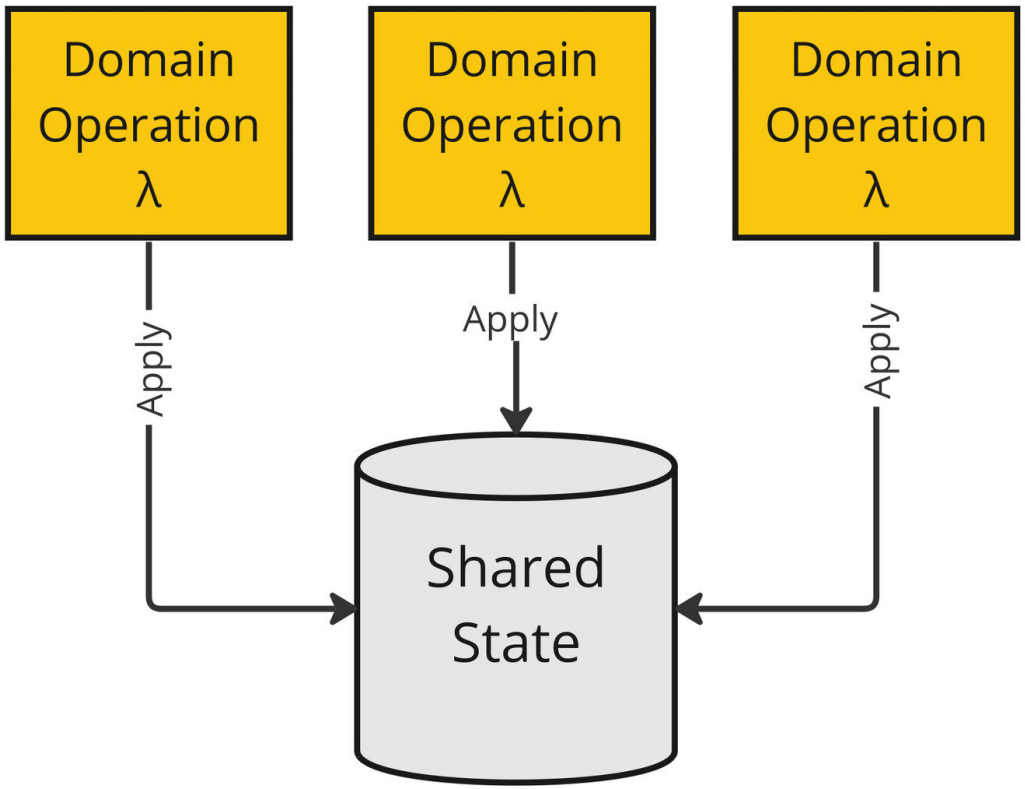
by Ville Vehviläinen | 23 Sep 2020

Microservices • Clojure • Development • Architecture

https://dev.solita.fi/2020/09/23/centralised-schema-management-microservices.html

Principle 1	Shared State
Principle 2	**Domain Operations**
Principle 3	Data Consistency
Principle 4	API Only
Principle 5	Access Control
Principle 6	Event Logging
Principle 7	Bitemporality

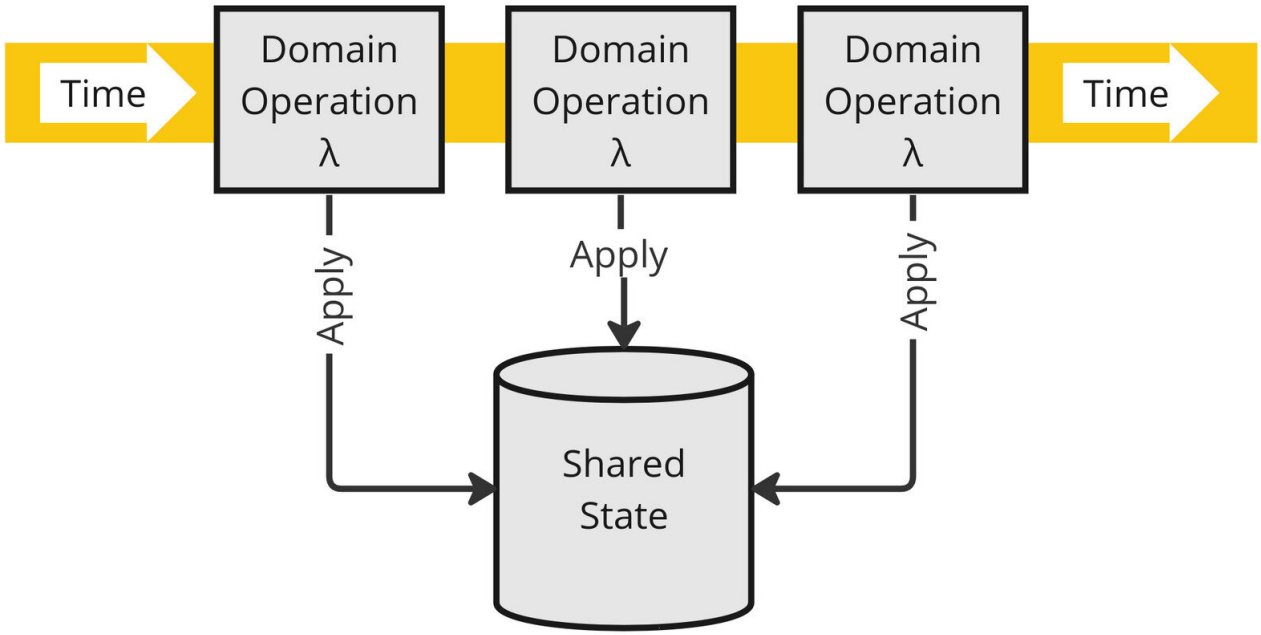
Domain Operations



update-customer-profile
confirm-payment
check-inventory-levels
modify-order-status
verify-account
schedule-delivery
reset-password
issue-ticket
track-shipment
update-order
create-invoice
adjust-billing
process-return
initiate-transfer
validate-membership
escalate-issue
generate-report
review-application
authorize-transaction
notify-user

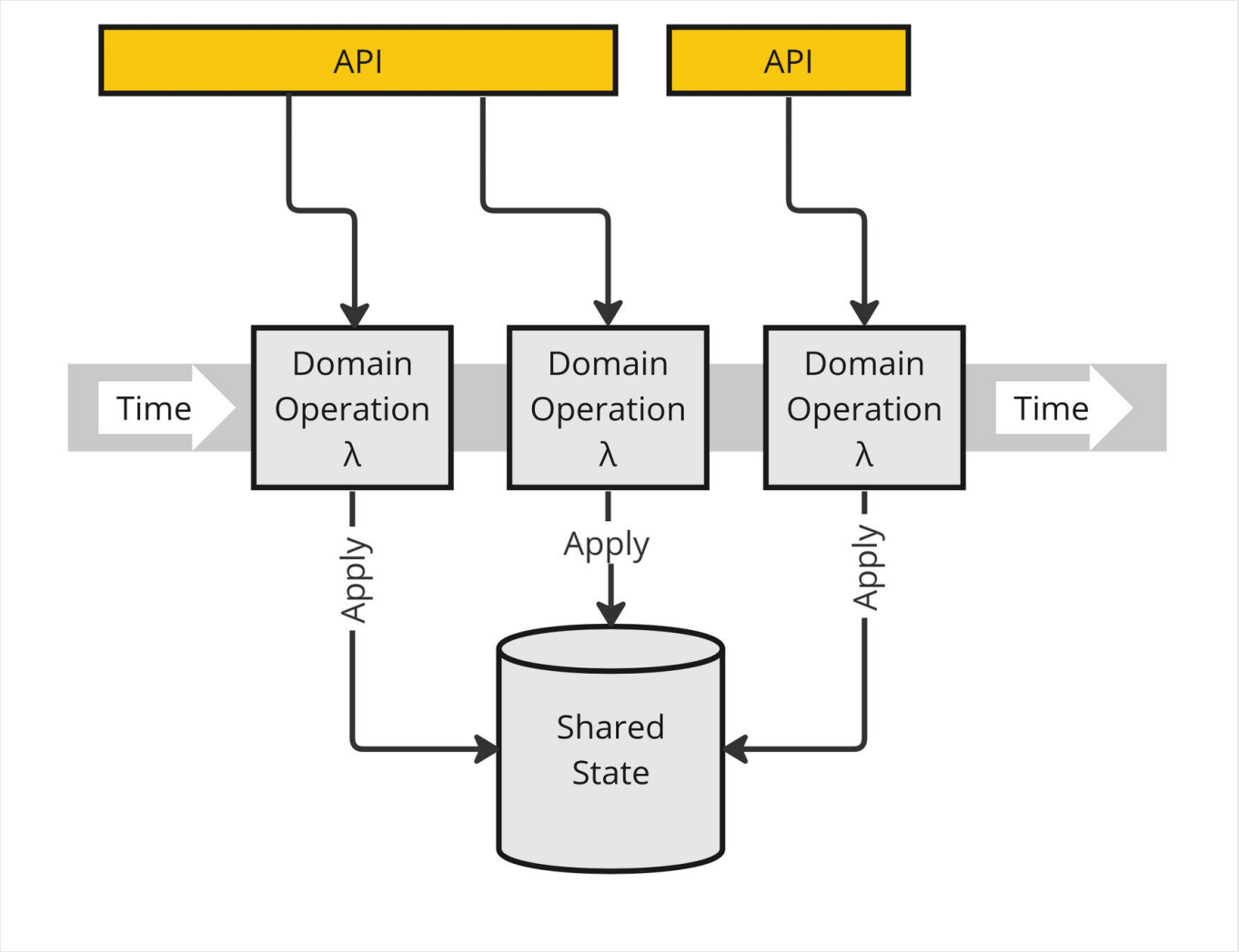
Principle 1	Shared State
Principle 2	Domain Operations
Principle 3	**Data Consistency**
Principle 4	API Only
Principle 5	Access Control
Principle 6	Event Logging
Principle 7	Bitemporality

Data Consistency



Principle 1	Shared State
Principle 2	Domain Operations
Principle 3	Data Consistency
Principle 4	**API Only**
Principle 5	Access Control
Principle 6	Event Logging
Principle 7	Bitemporality

API Only


API
API
Domain Operation λ
Domain Operation λ
Domain Operation λ
Time
Time
Apply
Apply
Apply
Shared State

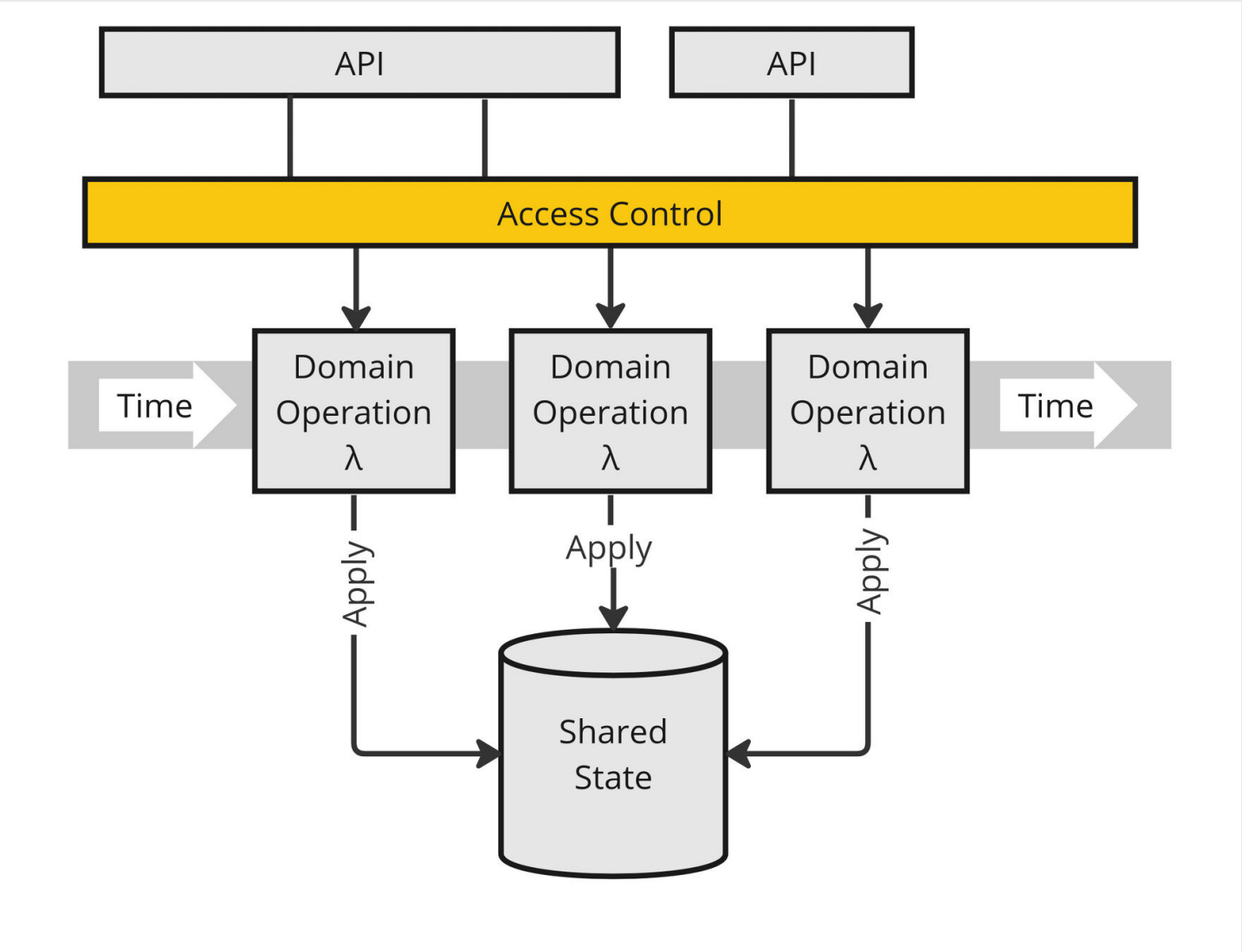

#GP2025
Meeting & Work Pods
From £5.99 Per Hour
1. Locate
Download the BookAPod App
2. Book
Select date, time & pod you want.
Pod as you go. No contracts or subscriptions.
3. Access
Unlock your pod via the BookAPod app.
Share access to trusted others with your eKey.
Parking Reminder
M1


#GP2025
Meeting & Work Pods
From £5.99 Per Hour
1. Locate
Download the BookAPod App
BOOK NOW
2. Book
Select date, time & pod you want.
Pod as you go. No contracts or subscriptions.
3. Access
Unlock your pod via the BookAPod app.
Share access to trusted others with your eKey.


OAUTH
2
OAUTH

Principle 1	Shared State
Principle 2	Domain Operations
Principle 3	Data Consistency
Principle 4	API Only
Principle 5	**Access Control**
Principle 6	Event Logging
Principle 7	Bitemporality

Access Control



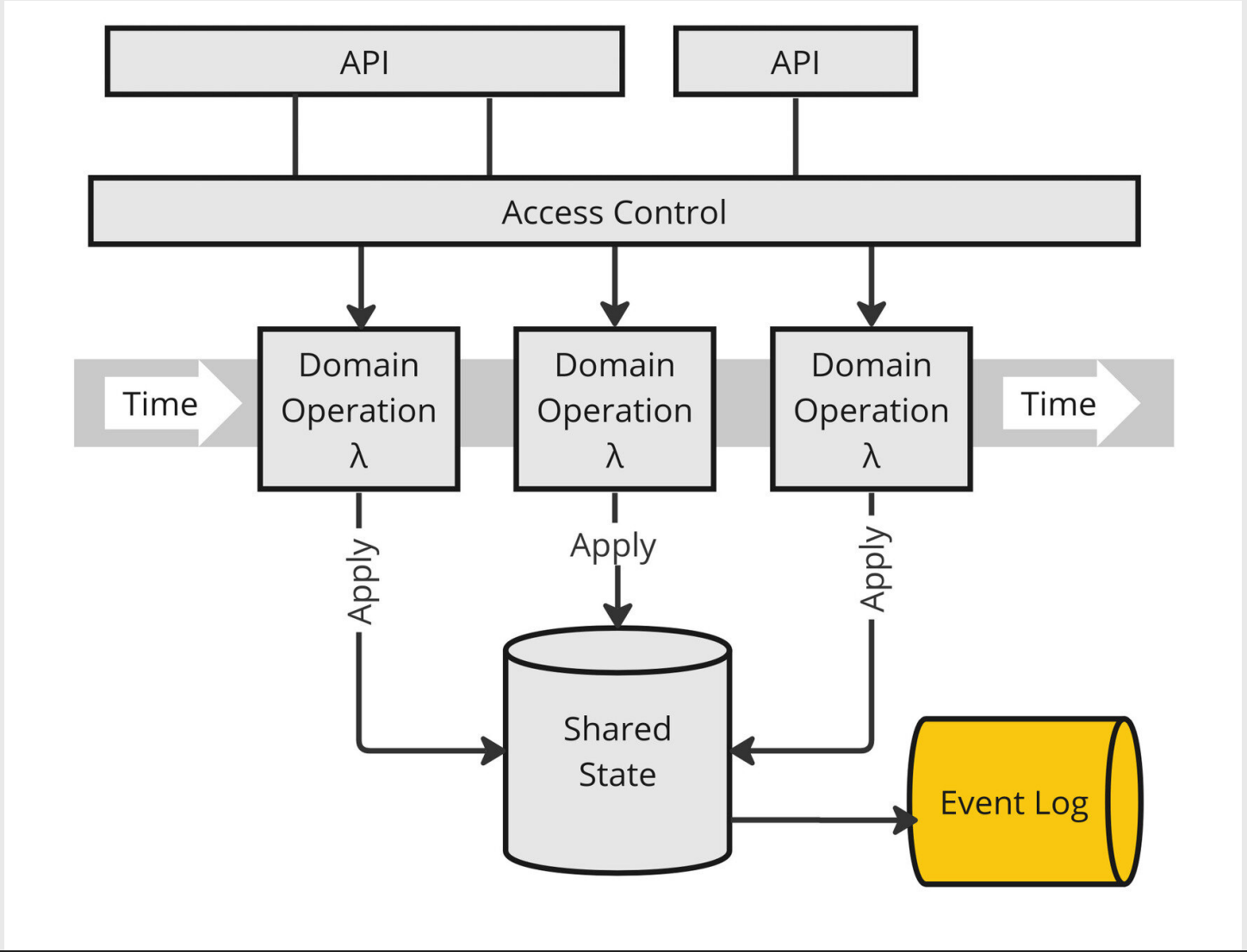
Access Control

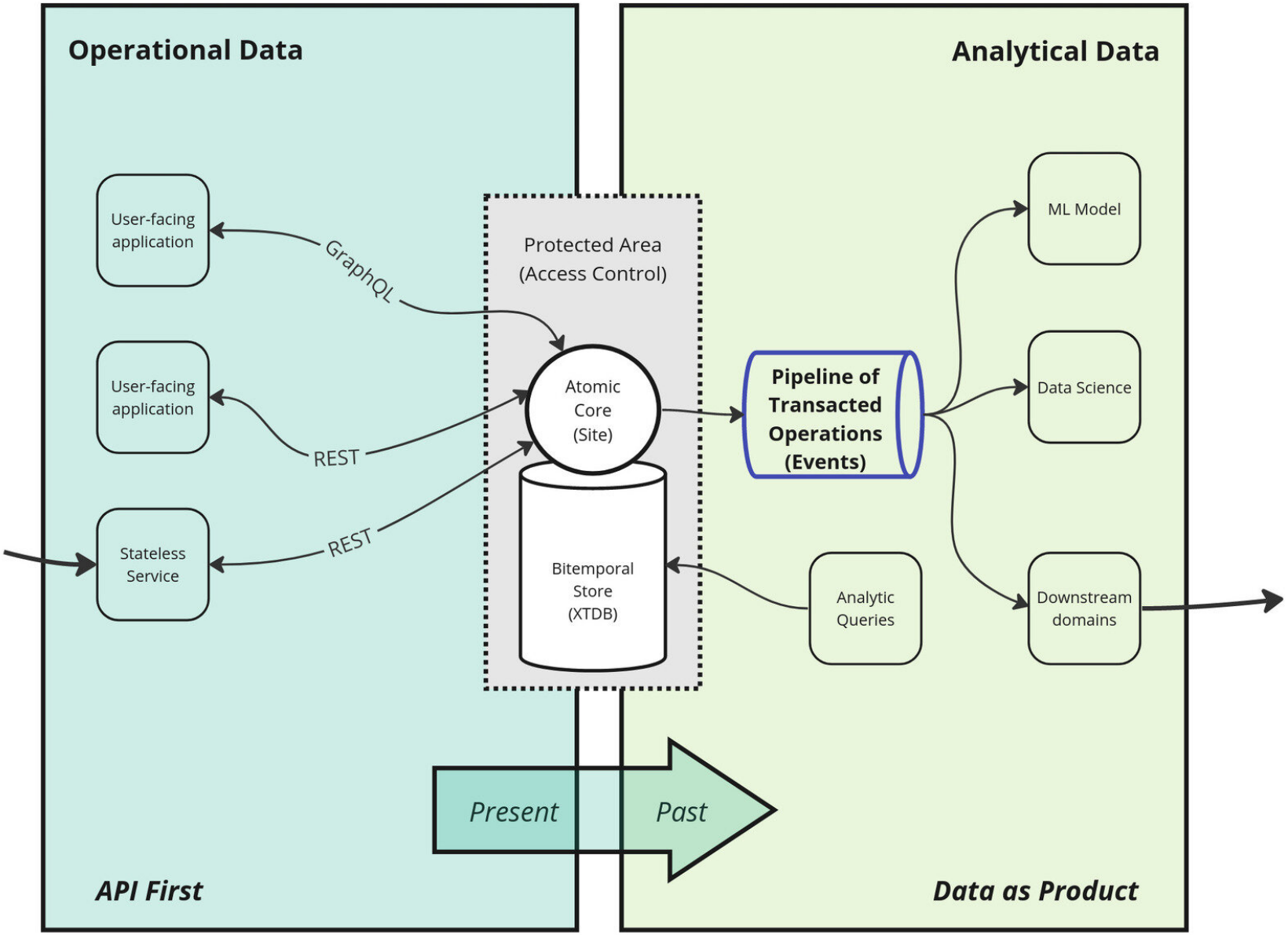
“ Executive Order 14028, “Improving the Nation’s Cybersecurity”, pushes agencies to adopt zero trust cybersecurity principles and adjust their network architectures accordingly.

https://zerotrust.cyber.gov/zero-trust-maturity-model/ May 12, 2021

Principle 1	Shared State
Principle 2	Domain Operations
Principle 3	Data Consistency
Principle 4	API Only
Principle 5	Access Control
Principle 6	**Event Logging**
Principle 7	Bitemporality

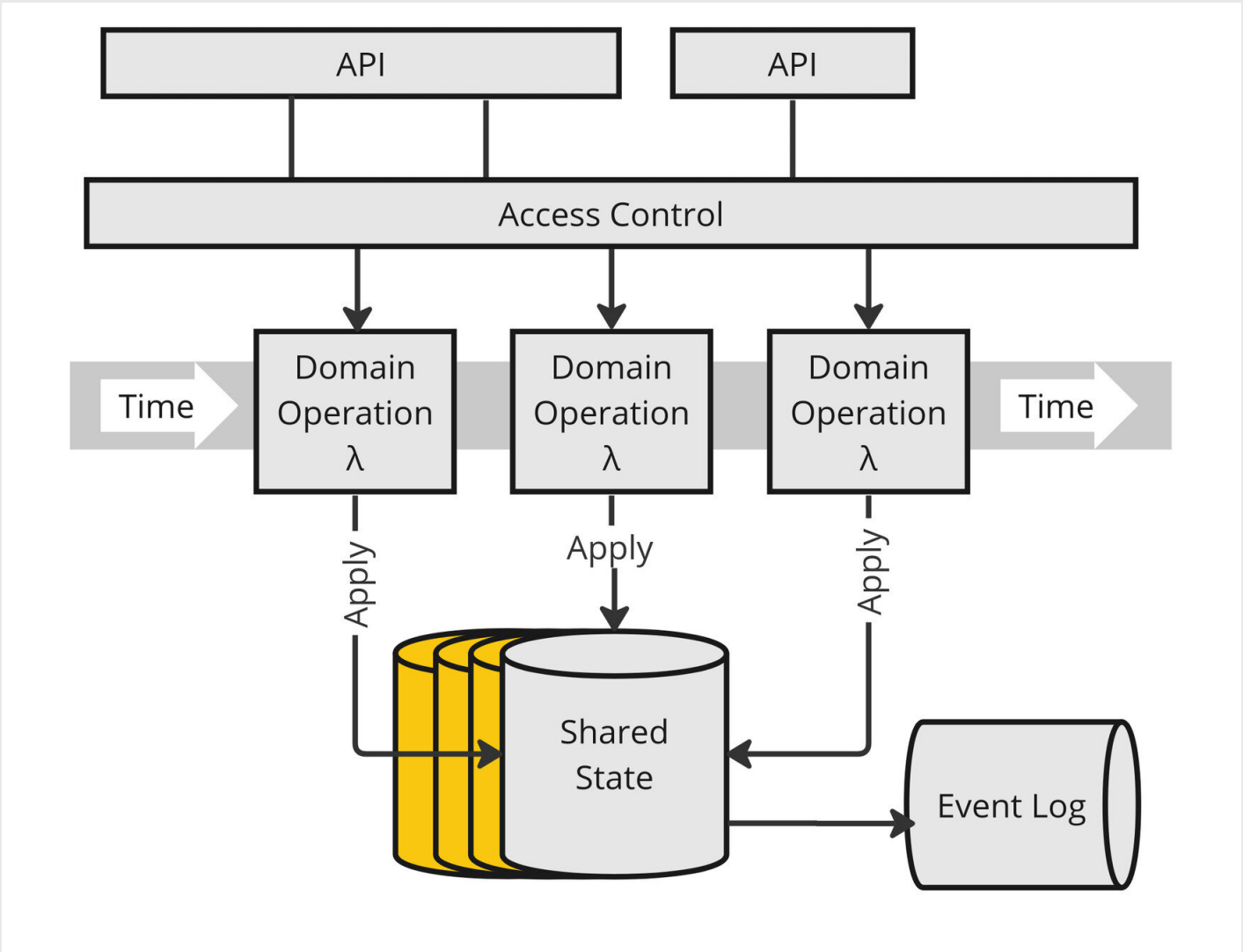
Event Logging



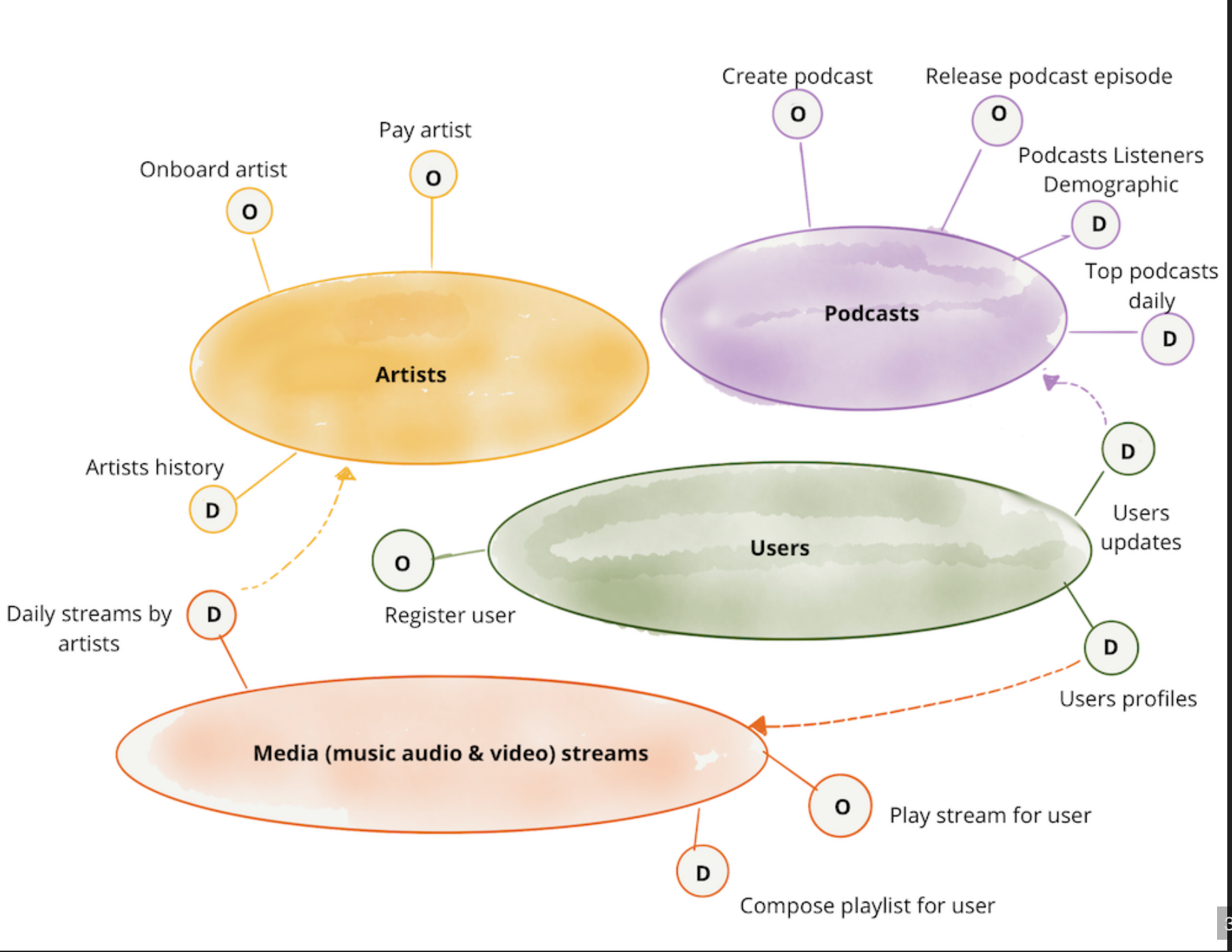

Operational Data
Analytical Data
User-facing application
GraphQL
User-facing application
REST
Stateless Service
REST
Protected Area (Access Control)
Atomic Core (Site)
Bitemporal Store (XTDB)
Pipeline of Transacted Operations (Events)
ML Model
Data Science
Analytic Queries
Downstream domains
Present
Past
API First
Data as Product

Principle 1	Shared State
Principle 2	Domain Operations
Principle 3	Data Consistency
Principle 4	API Only
Principle 5	Access Control
Principle 6	Event Logging
Principle 7	**Bitemporality**

Bitemporality



Wrap Up


Create podcast
O
Release podcast episode
O
Pay artist
Onboard artist
O
O
Podcasts Listeners Demographic
D
Top podcasts daily
D
Podcasts
Artists
Artists history
D
D
Users updates
Users
O
Register user
Daily streams by artists
D
D
Users profiles
Media (music audio & video) streams
O
Play stream for user
D
Compose playlist for user

Recommendations

- Evaluate moving state from services to a central store: "stateless microservices"
- Adopt OpenAPI
- Adopt OAuth2
- Evaluate GraphQL
- Evaluate JSON-LD
- Buy a copy of 'Data Mesh'

Questions?

Malcolm Sparks, CTO@JUXT

mal@juxt.pro

@malcolmsparks (GitHub, X, LinkedIn, etc.)